AFRL-AFOSR-VA-TR-2016-0329

AFOSR BRI: Co-Design of Hardware/Software for Predicting MAV Aerodynamics

Wu-Chun Feng
VIRGINIA POLYTECHNIC INST AND STATE UNIVERSITY
300 TURNER ST NW, SUITE 4200
BLACKSBURG, VA 24061-0001

09/27/2016
Final Report

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/RTA2

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01-03-2016	Final	01-September-2012 to 31-October-2015

4. TITLE AND SUBTITLE
AFOSR BRI: Co-Design of Hardware/Software for Predicting MAV Aerodynamics

5a. CONTRACT NUMBER
FA9550-12-1-0442

5b. GRANT NUMBER
BAA-AFOSR-2012-0001

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Feng, Wu-Chun

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
VIRGINIA POLYTECHNIC INSTITUTE & STATE UNIVERSITY
VIRGINIA TECH
1880 PRATT DR STE 2006
BLACKSBURG VA 24060-3580

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
USAF, AFRL DUNS 143574726
AF OFFICE OF SCIENTIFIC RESEARCH
875 N. RANDOLPH ST. ROOM 3112 ARLINGTON VA 22203
DANIEL P. SMITH 703-588-8494

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR BRI

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION A: Distribution approved for public release.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
While Moore's Law theoretically doubles processor performance every 24 months, much of the realizable performance remains untapped because the burden falls to the (less informed) domain scientist or engineer to exploit parallel hardware for performance gains. Even when such untapped hardware potential is fully realized, it is often not coupled with advances in algorithmic innovation, which can deliver further (multiplicative) speed-up beyond Moore's Law, as noted in the AFOSR BAA. In this project, we propose a formal co-design process for the structured grid and unstructured grid motifs found in computational fluid dynamics (CFD) in support of aerodynamic predictions for micro-air vehicles (MAVs). While many past efforts to develop such CFD codes on accelerated processors showed limited success, our hardware/software co-design approach created malleable algorithms that could be mapped and optimized onto the right type of processing core at the right time, and in turn, deliver an order-of-magnitude better performance than would have otherwise been possible by Moore's Law alone.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Wu-chun Feng
U	U	U	UU		19b. TELEPHONE NUMBER *(Include area code)* +1-540-231-1192

DISTRIBUTION A: Distribution approved for public release.

INSTRUCTIONS FOR COMPLETING SF 298

1. REPORT DATE. Full publication date, including day, month, if available. Must cite at least the year and be Year 2000 compliant, e.g. 30-06-1998; xx-06-1998; xx-xx-1998.

2. REPORT TYPE. State the type of report, such as final, technical, interim, memorandum, master's thesis, progress, quarterly, research, special, group study, etc.

3. DATES COVERED. Indicate the time during which the work was performed and the report was written, e.g., Jun 1997 - Jun 1998; 1-10 Jun 1996; May - Nov 1998; Nov 1998.

4. TITLE. Enter title and subtitle with volume number and part number, if applicable. On classified documents, enter the title classification in parentheses.

5a. CONTRACT NUMBER. Enter all contract numbers as they appear in the report, e.g. F33615-86-C-5169.

5b. GRANT NUMBER. Enter all grant numbers as they appear in the report, e.g. AFOSR-82-1234.

5c. PROGRAM ELEMENT NUMBER. Enter all program element numbers as they appear in the report, e.g. 61101A.

5d. PROJECT NUMBER. Enter all project numbers as they appear in the report, e.g. 1F665702D1257; ILIR.

5e. TASK NUMBER. Enter all task numbers as they appear in the report, e.g. 05; RF0330201; T4112.

5f. WORK UNIT NUMBER. Enter all work unit numbers as they appear in the report, e.g. 001; AFAPL30480105.

6. AUTHOR(S). Enter name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. The form of entry is the last name, first name, middle initial, and additional qualifiers separated by commas, e.g. Smith, Richard, J, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES). Self-explanatory.

8. PERFORMING ORGANIZATION REPORT NUMBER. Enter all unique alphanumeric report numbers assigned by the performing organization, e.g. BRL-1234; AFWL-TR-85-4017-Vol-21-PT-2.

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES). Enter the name and address of the organization(s) financially responsible for and monitoring the work.

10. SPONSOR/MONITOR'S ACRONYM(S). Enter, if available, e.g. BRL, ARDEC, NADC.

11. SPONSOR/MONITOR'S REPORT NUMBER(S). Enter report number as assigned by the sponsoring/ monitoring agency, if available, e.g. BRL-TR-829; -215.

12. DISTRIBUTION/AVAILABILITY STATEMENT. Use agency-mandated availability statements to indicate the public availability or distribution limitations of the report. If additional limitations/ restrictions or special markings are indicated, follow agency authorization procedures, e.g. RD/FRD, PROPIN, ITAR, etc. Include copyright information.

13. SUPPLEMENTARY NOTES. Enter information not included elsewhere such as: prepared in cooperation with; translation of; report supersedes; old edition number, etc.

14. ABSTRACT. A brief (approximately 200 words) factual summary of the most significant information.

15. SUBJECT TERMS. Key words or phrases identifying major concepts in the report.

16. SECURITY CLASSIFICATION. Enter security classification in accordance with security classification regulations, e.g. U, C, S, etc. If this form contains classified information, stamp classification level on the top and bottom of this page.

17. LIMITATION OF ABSTRACT. This block must be completed to assign a distribution limitation to the abstract. Enter UU (Unclassified Unlimited) or SAR (Same as Report). An entry in this block is necessary if the abstract is to be limited.

AFOSR BRI: Co-Design of Hardware/Software for Predicting MAV Aerodynamics

PI: Wu-chun Feng, wfeng@vt.edu, +1-540-231-1192
Dept. of Computer Science
Virginia Tech, Blacksburg, Virginia

PI's Students Funded under this Grant:
Sriram Chivukula, MS (2014)
Xuewen ("Harry") Cui, PhD (expected 2018)
Islam Harb, PhD (expected 2019)
Kaixi Hou, PhD (expected 2017)
Paul Sathre, MS (2013) → Post-MS Researcher (2014-now)
Thomas Scogland, PhD (2015)

While Moore's Law theoretically doubles processor performance every 24 months, much of the realizable performance remains untapped because the burden falls to the (less informed) domain scientist or engineer to exploit parallel hardware for performance gains. Even when such untapped hardware potential is fully realized, it is often not coupled with advances in algorithmic innovation, which can deliver further (multiplicative) speed-up beyond Moore's Law, as noted in the AFOSR BAA. For example, in a heterogeneous system containing a CPU and GPU, a straightforward 1600-core GPU parallelization of a CPU-based n-body code for molecular modeling resulted in only an 88.4-fold speed-up over a serial, but SSE-vectorized, CPU code. An additional 4.2-fold was extracted when applying architecture-aware GPU optimizations, resulting in a 371-fold speed-up. By also leveraging algorithmic innovation via a hierarchical charge partitioning algorithm, we delivered an additional 216-fold speed-up, resulting in a multiplicative speed-up of 80,000-fold.

Therefore, in this project, we formalized the aforementioned co-design process for the n-body computational motif and adapted and applied it to the structured grid and unstructured grid motifs found in computational fluid dynamics (CFD) in support of aerodynamic predictions for micro-air vehicles (MAVs). While many past efforts to develop such CFD codes on accelerated processors showed limited success, our hardware/software co-design approach created malleable algorithms that could be mapped and optimized onto the right type of processing core at the right time, and in turn, deliver an order-of-magnitude better performance than would have otherwise been possible by Moore's Law alone. To further enhance our co-design process, we engaged hardware vendors to support our effort, and in turn, our research has assisted in guiding their future hardware design, e.g., our GPU-accelerated HokieSpeed supercomputer.

Overview

Many past efforts to develop computational fluid dynamics (CFD) codes in heterogeneous computing systems that consist of accelerated processors, such as graphical processing units (GPUs), have demonstrated limited success. This predicament was due in part to the relatively naive mapping of traditional CPU-based algorithms onto accelerated processors like the GPU

rather than a synergistic hardware/software co-design approach that (1) re-factors the CPU-based CFD algorithms and co-designs them to accelerated processors or (2) starts from first principles that underlie CFD, e.g., the fundamental computational motifs of structured/unstructured grids, and co-designs them in the context of these accelerated processors.

To achieve the desired metric(s) of success with respect to speed, programmability, portability, power consumption, energy efficiency, and combinations thereof, the different combinations of algorithm, software, and hardware need to be judiciously co-designed. In formalizing the co-design approach that we used for molecular modeling and applying it to CFD, we researched and developed CFD algorithms and supporting hardware for micro-air vehicle (MAV) simulations that achieved substantial speed-up over current simulations and provided significantly better hardware utilization. Such an increase in performance, realized via our co-design approach, allowed large eddy simulation (LES) calculations on 100-million element meshes to become routine. However, significant challenges existed in mapping specific codes designed for such simulations in heterogeneous computing environments, consisting of CPUs and GPUs (and very soon, accelerated processing units or APUs that combine the CPU and GPU onto the same processor die).

The high-performance simulation of MAVs required a number of performance-related choices at the *method* and *algorithm* levels that were co-designed with the underlying systems software (e.g., parallel libraries and run-time system) and hardware. In turn, system software and hardware choices needed to optimally support the algorithmic requirements of MAV simulation. As an example, consider a simulation that requires the solution of the unsteady Reynolds-Averaged Navier-Stokes equations (RANS) using the pressure projection approach, where we have a choice between structured or unstructured finite-element meshes and explicit or implicit time steps. For this scenario, we used a mesh-partitioning algorithm to provide a suitable domain decomposition. The subdomains were then distributed over the nodes, such that the mapping respected proximity of adjoining subdomains and with multiple subdomains per node, allowing subdomains to be reassigned dynamically for load balancing.

The pressure-projection approach requires the solution of a linear system (Poisson equation) for the pressure in each time step, even in the case of explicit time steps. Here, we assumed the problem was sufficiently large that iterative solvers were required. An implicit time step required the solution of a fully coupled nonlinear system of equations. Assuming a Newton-based approach, this involved computing a sequence of Jacobian matrices and solving the associated linear systems (which are larger than for the explicit case). Hence, we created efficient linear solvers towards delivering high performance in CFD simulations.

Each iteration of a linear solver involves one sparse matrix-vector product (matvec) and preconditioner-vector product (precvec, often with similar characteristics as the sparse matrix-vector product), multiple dot products, and multiple vector updates. (Note: The pattern of computation and communication captured by this linear solver is referred to as the *sparse linear algebra* motif or dwarf).

The main performance issues involved (1) reducing the cost of communication and synchronization between nodes, (2) load balancing (both computation and communication) between nodes and between the processors on a node, (3) keeping the number of iterations low

while using an efficient parallel preconditioner, (4) high (local) performance on (accelerated) processors for matvec, precvec, and dot product, and (5) vector update. While we achieved many improvements via modest modifications of standard algorithms. We also considered hardware choices that support the efficient implementation of such modifications, including the reduction of global communication and synchronization in inner products.

While the use of unstructured meshes, leading to irregular memory access, has little impact on the communication cost between nodes, it results in a significant impact on the efficient use of the processors and caches, particularly GPUs. Hence, high performance (by itself) might favor the choice of structured meshes.

Both for structured and unstructured meshes, the sparse matvec has very low flops per data fetch, making the operation typically memory bound. Various algorithmic transformations can make the use of unstructured meshes more efficient. Creating many relatively small subdomains and (re)ordering unknowns by subdomain led to better locality. In addition, using appropriate matrix data structures (e.g., Cray/Ellpack storage) and explicit prefetching of data significantly improved performance for general sparse matvecs. The (vectorized) matvec for structured meshes can be implemented efficiently by diagonals, possibly with some blocking. Improving the low number of flops per data item is more complicated, but it can be improved by various algorithmic modifications, such as combining S subsequent iterations (S times the work but (most) matrix elements must be fetched only once).

Finally, we produced a range of solver improvements, such as Krylov subspace recycling and updating preconditioners, that significantly reduced the number of iterations while introducing modest overhead that, in addition, had better performance characteristics on parallel, multi-core machines. For example, the overhead in Krylov subspace recycling involves a matvec with multiple vectors at once, and the orthogonalization of (a group of) vectors against a group of orthogonal vectors. Krylov subspace recycling will be particularly relevant to the fully implicit approach requiring the solution of large nonlinear systems of equations.

The relevant motifs above are sparse linear algebra, structured or unstructured grids, and dense linear algebra for small (relative to the total problem) local problems. *Co-designing motifs, supported by the run-time system, and modified solvers led to significant performance improvements and easier programming at the same time*.

Some of the algorithmic modifications above that influenced hardware choices included combining multiple iterations of the linear solvers, Krylov subspace recycling, and special implementations of the sparse matvec for matrices from unstructured meshes. We considered hardware that supports loading multiple (sub)vectors and efficient multiplication by a single (sub)matrix, can store/cache relatively large amounts of data, has relatively high bandwidth to cache or memory, and has high bandwidth access to accelerator. In turn, we optimized our algorithms to improve flops per data item fetched and favored algorithms and methods that allowed modifications that increased this ratio without introducing too much overhead.

Important co-design parameters included (1) the size and ordering of subdomains/patches; (2) size of sub- domain buffer (ghost) space (for message aggregation and longer local computations); (3) block sizes in algorithms and underlying data structures tuned to local/fast

memories. Blocks can correspond to finite- element method (FEM) elements, but can be smaller if needed; and (4) number of concurrent matvecs (possibly with submatrices).

In all, the research from this three-year grant resulted in more than 50 publications. A brief listing of our research innovations is provided below.

- Evolution of four distinctive GPU-accelerated CFD codes: GENIDLEST, SENSEI, RDGFLOW, and INCOMP3D, of which two achieved an order-of-magnitude speed-up.
- Detailed characterization of the programmability and performance challenges faced by each of the four CFD codes on a GPU, e.g.,
    - Memory boundedness and limited memory
    - Inter-cell and intra-cell dependency
    - Excessive branching
    - Mixed computational granularity
    - Optimization of common computational and communication idioms, i.e., dwarfs or motifs, such as ghost-cell packing
    - Co-design of hardware, software, and algorithm simultaneously, e.g., how to design a linear solver algorithm and realize it in software while keeping in mind the strengths of the underlying hardware architecture. Some examples include a block-sparse linear solver.
- Characterization of performance relative to programming language, e.g., CUDA vs. OpenACC. For this particular set of CFD codes, OpenACC codes ran up to twice as slow as their CUDA equivalents.
- Rigorous verification of the correctness of GPU-accelerated CFD codes.
- Maturation of newly researched and developed numerical methods from 2014 for many-core GPU environments, e.g., minimizing execution time via fewer expensive iterations versus more cheap iterations, GPU-parallelized Rosenbrock-Krylov method, and integrated co-design of CUDA-parallelized solvers with CFD.
- New numerical methods, including block-orthogonal Rosenbrock-Krylov (ROK) / Exponential Krylov (EXPK) methods, linearly implicit Runge-Kutta W (LIRK-W), and implicit-explicit general linear methods.
- Realization of a novel memory analysis tool that can be used to project the efficacy of porting a code from CPU to GPU.
- Abstraction of additional common data structures and algorithmic dwarfs (or motifs) and their optimization and incorporation into an accelerator-based library framework called MetaMorph, which for the first time anywhere enables different accelerator devices to interoperate, e.g., a program simultaneously run on an AMD GPU, NVIDIA GPU, and Intel Xeon Phi.
- Creation of a prototypical runtime system called CoreTSAR, short for Core Task-Size Adapting Runtime, that automatically distributes tasks across CPUs and GPUs to execute simultaneously. (This is in contrast to the bulk- synchronous parallel style of execution that has been traditionally adopted by the high-performance computing community, where execution alternates between the CPU and GPU.)

AFOSR Deliverables Submission Survey

Response ID:6911 Data

1.

Report Type

Final Report

Primary Contact Email

Contact email if there is a problem with the report.

wfeng@vt.edu

Primary Contact Phone Number

Contact phone number if there is a problem with the report

+1-540-231-1192

Organization / Institution name

Virginia Tech

Grant/Contract Title

The full title of the funded effort.

AFOSR BRI: Co-Design of Hardware/Software for Predicting MAV Aerodynamics

Grant/Contract Number

AFOSR assigned control number. It must begin with "FA9550" or "F49620" or "FA2386".

FA9550-12-1-0442

Principal Investigator Name

The full name of the principal investigator on the grant or contract.

Wu-chun Feng

Program Officer

The AFOSR Program Officer currently assigned to the award

Jean-Luc Cambier

Reporting Period Start Date

09/01/2012

Reporting Period End Date

10/31/2015

Abstract

While Moore's Law theoretically doubles processor performance every 24 months, much of the realizable performance remains untapped because the burden falls to the (less informed) domain scientist or engineer to exploit parallel hardware for performance gains. Even when such untapped hardware potential is fully realized, it is often not coupled with advances in algorithmic innovation, which can deliver further (multiplicative) speed-up beyond Moore's Law, as noted in the AFOSR BAA. In this project, we propose a formal co-design process for the structured grid and unstructured grid motifs found in computational fluid dynamics (CFD) in support of aerodynamic predictions for micro-air vehicles (MAVs). While many past efforts to develop such CFD codes on accelerated processors showed limited success, our hardware/software co-design approach created malleable algorithms that could be mapped and optimized onto the right type of processing core at the right time, and in turn, deliver an order-of-magnitude better performance than would have otherwise been possible by Moore's Law alone.

Distribution Statement

This is block 12 on the SF298 form.

Distribution A - Approved for Public Release

Explanation for Distribution Statement

If this is not approved for public release, please provide a short explanation. E.g., contains proprietary information.

SF298 Form

Please attach your SF298 form. A blank SF298 can be found here. Please do not password protect or secure the PDF The maximum file size for an SF298 is 50MB.

Upload the Report Document. File must be a PDF. Please do not password protect or secure the PDF . The maximum file size for the Report Document is 50MB.

Upload a Report Document, if any. The maximum file size for the Report Document is 50MB.

Archival Publications (published) during reporting period:

Theses

N. Balasubramanian, "ScalaMemAnalysis: Compositional Cache Analysis of Compressed Memory Traces," M.S. Thesis, Department of Computer Science, North Carolina State University, Advisor: F. Mueller, 2014.

A. Grim McNally, "Reusing and Updating Preconditioners for Sequences of Matrices," M.S. Thesis, Department of Mathematics, Virginia Tech, Advisor: Eric de Sturler, 2015.

T. Scogland, "Runtime Adaptation for Autonomic Heterogeneous Computing," Ph.D. Thesis, Department of Computer Science, Virginia Tech, Advisor: Wu-chun Feng, 2014.

2013

E.M. Constantinescu, A. Sandu, "Extrapolated Multirate Methods for Differential Equations with Multiple Time Scales," Journal of Scientific Computing, 56(1):28-44, 2013.

A. Sandu, A. St-Cyr, "Stability Analysis of the Matrix-Free Linearly Implicit Euler Method," Domain Decomposition Methods in Science and Engineering, R. Bank, M. Holst, O. Widlund, and J. Xu editors, Lecture Notes in Computational Science and Engineering, 91:403-410, 2013.

A. Sandu, "Two Classes of Implicit-Explicit General Linear Methods," Auckland Numerical Ordinary Differential Equations Conference (ANODE 2013), Auckland, New Zealand, Jan. 2013.

P. Tranquilli, A. Sandu, "Matrix-free Integrators for Large Discretized PDEs," SIAM Conference on Computational Science and Engineering, Boston, Massachusetts, Feb.-Mar. 2013.

J.M. Derlaga, T.S. Phillips, C.J. Roy, "SENSEI Computational Fluid Dynamics Code: A Case Study in Modern Fortran Software Development," 21st AIAA Computational Fluid Dynamics Conference, San Diego, Jun. 2013.

M. Guenther, A. Sandu et al., "Nonlinear Stability of Generalized Additive and Partitioned Implicit Multirate Runge-Kutta Schemes," SciCADE, Valladolid, Spain, Sept. 2013.

P. Tranquilli, A. Sandu, "Rosenbrock-Krylov Time Stepping Methods," SciCADE, Valldolid, Spain, Sept. 2013.

H. Zhang, A. Sandu et al., "A Class of Implicit-Explicit General Linear Methods," SciCADE, Valladolid, Spain, Sept. 2013.

S. R. Glandon, P. Tranquilli, A. Sandu, "Acceleration of Matrix-Free Time Integration Methods," Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems (ScalA) at SC13, Nov. 2013.

2014

A. Cardone, Z. Jackiewicz, A. Sandu, and H. Zhang: "Extrapolation-Based Implicit-Explicit General Linear Methods", Numerical Algorithms, 65(3):377-399, 2014.

A. Cardone, Z. Jackiewicz, A. Sandu, and H. Zhang, "Extrapolated IMEX Runge-Kutta Methods," Mathematical Modelling and Analysis, 19(1):18-43, 2014.

L. Luo, J. R. Edwards, H. Luo, F. Mueller, "GPU port of a parallel incompressible Navier-Stokes solver based on OpenACC and MVAPICH2," AIAA Aviation and Aeronautics Forum and Exposition, Atlanta, GA, 2014.

P. Tranquilli and A. Sandu, "Exponential-Krylov Methods for Ordinary Differential Equations," Journal of Computational Physics, 278(C):31-46, 2014.

H. Zhang, A. Sandu, S. Blaise, "Partitioned and Implicit-Explicit General Linear Methods for ODEs," Journal on Scientific Computing, 61(1):119-144, 2014.
B. Pickering, C. Jackson, T. Scogland, W. Feng, C. Roy, "Directive-Based GPU Programming for Computational Fluid Dynamics," 52nd AIAA Aerospace Sciences Meeting (SciTech), National Harbor, MD, Jan. 2014.
Y. Xia, L. Luo, H. Luo, J. Edwards, F. Mueller, "GPU Acceleration of a Reconstructed Discontinuous Galerkin Method for Compressible Flows on Unstructured Grids," 52nd AIAA Aerospace Sciences Meeting (SciTech), National Harbor, MD, Jan. 2014.
Y. Xia, H. Luo, L. Luo, J. Edwards, J. Lou, F. Mueller, "OpenACC-based GPU Acceleration of a 3-D Unstructured Discontinuous Galerkin Method," 52nd AIAA Aerospace Sciences Meeting (SciTech), National Harbor, MD, Jan. 2014.
L. Luo, J. Edwards, H. Luo, F. Mueller, "Performance Assessment of Multi-block LES Simulations using Directive- based GPU Computation in a Cluster Environment," 52nd AIAA Aerospace Sciences Meeting (SciTech), National Harbor, MD, Jan. 2014.
C. Li, Y. Yang, H. Dai, S. Yan, F. Mueller, H. Zhou, "Understanding the Tradeoffs between Software-Managed vs. Hardware-Managed Caches in GPUs," IEEE International Symposium on Performance Analysis of Systems and Software, Mar. 2014.
T. Scogland, W. Feng, B. Rountree, B. de Supinski, "CoreTSAR: Adaptive Worksharing for Heterogeneous Systems," Int'l Supercomputing Conf., Leipzig, Germany, Jun. 2014
L. Luo, J. Edwards, H. Luo, F. Mueller, "GPU Port of a Parallel Incompressible Navier-Stokes Solver based on OpenACC and MVAPICH2," AIAA Aviation 2014, Atlanta, GA, Jun. 2014.
Y. Xia, L. Luo, H. Luo, J. Lou, J. Edwards, F. Mueller, "On the Multi-GPU Computing of a Reconstructed Discontinuous Galerkin Method for Compressible Flows on 3D Hybrid Grids," 7th AIAA Theoretical Fluid Mechanics Conference, Atlanta, GA, Jun. 2014.
A. Amritkar, D.Tafti, P. Sathre, K. Hou, S. Chivakula, W. Feng. "Accelerating Bio-Inspired MAV Computations using GPUs." AIAA Aviation and Aeronautics Forum and Exposition 2014, Atlanta, GA, Jun. 2014.
K. Swirydowicz, E. de Sturler, X. Xu, C. Roy, "Fast Solvers and Preconditioners," SIAM Annual Meeting, Chicago, IL, Jul. 2014.
A. Cardone, Z. Jackiewicz, A. Sandu, and H. Zhang, "Implicit-Explicit Multivalue Methods for semi-discretized PDEs," 10th AIMS Conference on Dynamical Systems Differential Equations and Applications, Madrid, Spain, Jul. 2014.
M. Guenther, C. Hachtel, and A. Sandu, "Multirate GARK Schemes for Multiphysics Problems, 10th International Conference on Scientific Computing in Electrical Engineering Wuppertal, Germany, 23:115-121, Jul. 2014.
A. Amritkar, D. Tafti. "CFD Computations using Preconditioned Krylov Solver on GPUs." ASME 2014 Fluids Engineering Division Summer Meeting, Chicago, IL, Aug. 2014.
K. Swirydowicz, A. Amritkar, E. de Sturler, D. Tafti. "Recycling Krylov Subspaces for CFD Application," ASME 2014 Fluids Engineering Division Summer Meeting, Chicago, IL, Aug. 2014.
H. Zhang, A. Sandu, "FATODE: A Library for Forward, Adjoint, and Tangent Linear Integration of ODEs," SIAM Journal on Scientific Computing, 36(5):504-523, 2014.
P. Tranquilli, A. Sandu, "Rosenbrock-Krylov Methods for Large Systems of Differential Equations," SIAM Journal on Scientific Computing, 36(3):1313-1338, 2014.
A. Cardone, Z. Jackiewicz, A. Sandu, H. Zhang, "Extrapolated IMEX Runge-Kutta Methods," Mathematical Modelling and Analysis, 19(1):18-43, 2014.
A. Cardone, Z. Jackiewicz, A. Sandu, H. Zhang, "Extrapolation-Based Implicit-Explicit General Linear Methods," Numerical Algorithms, 65(3):377-399, 2014.
H. Zhang, A. Sandu, "Application of implicit-explicit general linear methods to reaction-diffusion problems," 12th International Conference of Numerical Analysis and Applied Mathematics (ICNAAM 2014), Rhodes, Greece, Sept. 2014.

2015

A. Sandu, "Rosenbrock Methods with an Explicit First Stage for Mildly Stiff PDEs," International Journal of

Computer Mathematics, 93(6):995-1010, 2015.
A. Sandu, M. Guenther, "A Generalized-Structure View of Additive Runge-Kutta Methods," SIAM Journal on Numerical Analysis, 53(1):17-42, 2015.
E. Zharovsky, A. Sandu, H. Zhang: "A Class of Implicit-Explicit Two-step Runge-Kutta Methods," SIAM Journal on Numerical Analysis, 53(1):321-341, 2015.
L. Luo, J. R. Edwards, H. Luo, F. Mueller, "Advanced Optimizations of An Implicit Navier-Stokes Solver on GPGPU based on a Fine-grained BILU solver," 53rd AIAA Aerospace Sciences Meeting (SciTech), Kissimmee, FL, Jan. 2015.
J. Lou, Y. Xia, L. Luo, H. Luo, J. Edwards, F. Mueller, "OpenACC-Based GPU Acceleration of a p-multigrid Discontinuous Galerkin Method for Compressible Flows on 3D Unstructured Grids," 53rd AIAA Aerospace Sciences Meeting, Kissimmee, FL, Jan. 2015.
M. Guenther, A. Sandu, "Multirate GARK Schemes," Computational Techniques for Time Dependent Coupled Multiphysics and Multiscale Problems Mini-Symposium MS6, SIAM Conference on Computational Science and Engineering, Salt Lake City, Utah, Mar. 2015.
A. Sandu, M. Guenther, "Generalized Structure Additive Runge-Kutta Methods," Advances in Time-stepping Methods Mini-Symposium MS136, SIAM Conference on Computational Science and Engineering, Salt Lake City, Utah, Mar. 2015.
P. Tranquilli, A. Sandu, R. Glandon, "K-Methods, an Extension of Exponential and Rosenbrock Time Integrators," Advances in Time-stepping Methods Mini-Symposium MS136, SIAM Conference on Computational Science and Engineering, Salt Lake City, Utah, Mar. 2015.
Y. Xia, J. Lou, H. Luo, J. Edwards, and F. Mueller, "OpenACC Acceleration of an Unstructured CFD Solver Based on a Reconstructed Discontinuous Galerkin Method for Compressible Flows," International Journal for Numerical Methods in Fluids, 78(3):123-139, May 2015.
L. Luo, J. R. Edwards, H. Luo, F. Mueller, "Fine-grained Optimizations of Implicit Algorithms in An Incompressible Navier-Stokes Solver on GPGPU," AIAA Aviation and Aeronautics Forum and Exposition, Dallas, TX, Jun. 2015.
A. Amritkar, D. Tafti, "CFD Computations using a Preconditioned Krylov Solver on GPUs," Journal of Fluids Engineering, 138(1):011402, Jul. 2015.
L. Luo, J. R. Edwards, H. Luo, F. Mueller, "A Fine-grained Block ILU Scheme on Regular Structures for GPGPU," Computers & Fluids, 119:149-161, Sept. 2015.
H. Zhang, A. Sandu, and P. Tranquilli, "Application of Approximate Matrix Factorization to High-Order Linearly-Implicit Runge-Kutta Methods,",Journal of Computational and Applied Mathematics, 286(C):196-210, Oct. 2015.
T. Scogland, W. Feng, B. Rountree, and B. de Supinski, "CoreTSAR: Core Task-Size Adapting Runtime," IEEE Transactions on Parallel and Distributed Systems, 26(11):2970-2983, Nov. 2015.
A. Amritkar, E. de Sturler, K. Swirydowicz, D. Tafti, K. Ahuja, "Recycling Krylov Subspaces for CFD Applications," Journal of Computational Physics. 303(C):222-237, Dec. 2015.

2016
A. Grim McNally, E. de Sturler, S. Gugercin, "Preconditioned Parameterized Linear Systems," arXiv.org, Mar. 2016.
H. Zhang, A. Sandu, and S. Blaise, "High Order Implicit–Explicit General Linear Methods with Optimized Stability Regions," SIAM Journal on Scientific Computing, 38(3), A1430–A1453, May 2016.
M. Guenther, A. Sandu, "Multirate Generalized Additive Runge-Kutta Methods," Numerische Mathematik, 133(3):497-524, Jul. 2016.
A. Helal, P. Sathre, W. Feng, "MetaMorph: A Library Framework for Interoperable Kernels on Multi- and Many-core Clusters," SC|16: The IEEE/ACM International Conference for High Performance Computing, Networking, Storage and Analysis (Supercomputing), Salt Lake City, Utah, Nov. 2016.

New discoveries, inventions, or patent disclosures:
Do you have any discoveries, inventions, or patent disclosures to report for this period?

Yes

Please describe and include any notable dates

Do you plan to pursue a claim for personal or organizational intellectual property?

No

Changes in research objectives (if any):

Change in AFOSR Program Officer, if any:

From Dr. Fariba Fahroo to Dr. Jean-Luc Cambier

Extensions granted or milestones slipped, if any:

AFOSR LRIR Number

LRIR Title

Reporting Period

Laboratory Task Manager

Program Officer

Research Objectives

Technical Summary

Funding Summary by Cost Category (by FY, $K)

	Starting FY	FY+1	FY+2
Salary			
Equipment/Facilities			
Supplies			
Total			

Report Document

Report Document - Text Analysis

Report Document - Text Analysis

Appendix Documents

2. Thank You

E-mail user

Sep 21, 2016 04:27:52 Success: Email Sent to: wfeng@vt.edu